HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200312021-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): James A. Lamb, et al.

Confirmation No.: 2113

Application No.: 10/692,939

Examiner: Charles E. Anya

Filing Date: October 24, 2003

Group Art Unit: 2194

Title: PROGRAM INTERFACE ARCHITECTURE

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on April 2, 2008.

☒ The fee for filing this Appeal Brief is $510.00 (37 CFR 41.20).

☐ No Additional Fee Required.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120	☐ 2nd Month $460	☐ 3rd Month $1050	☐ 4th Month $1640

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $ 510. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

☒ A duplicate copy of this transmittal letter is enclosed.

☐ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450
Date of Deposit:

OR

☒ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.
Date of facsimile: 05/06/2008
Typed Name: Jennifer L. Vomhof
Signature:

Rev 10/07(AplBrief)

Respectfully submitted,
James A. Lamb, et al.
By

Edward J. Brooks III
Attorney/Agent for Applicant(s)
Reg No.: 40,925
Date: 05/06/2008
Telephone: (612) 236-0120

05/06/2008 VBUI11 00000025 082025 10692939
01 FC:1402 510.00 DA

Docket No.: 200312021-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.	:	10/692,939
Appellants:	:	James A. Lamb, et al.
Filed:	:	October 24, 2003
TC/A.U.	:	2194
Examiner:	:	Charles E. Anya
Title	:	PROGRAM INTERFACE ARCHITECTURE

APPEAL BRIEF

MS APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir or Madame:

This brief, in compliance with 37 C.F.R. § 41.37, is in furtherance of the Notice of Appeal filed under 37 C.F.R. § 41.31 on April 2, 2008.

This brief is accompanied by the fee set forth in 37 CFR § 41.20(b)(2), as described in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37:

I. Real Party In Interest

II. Related Appeals and Interferences

III. Status of Claims

IV. Status of Amendments

V. Summary of Claimed Subject Matter

VI. Grounds of Rejection to be Reviewed on Appeal

VII. Argument

VIII. Claims Appendix

IX. Evidence Appendix

X. Related Proceedings Appendix

The final page of this brief bears the attorney's signature.

I. REAL PARTY IN INTEREST

The real parties in interest for this appeal are:

A. The Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC"); and

B. HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. RELATED APPEALS AND INTERFERANCES

Appellant submits that no related application is presently undergoing appeal or interference proceedings.

III. STATUS OF CLAIMS

A. Total Claims: 1-32

B. Current Status of Claims:

1. Claims canceled: 12, 13
2. Claims withdrawn: none
3. Claims pending: 1-11, 14-32
4. Claims allowed: none
5. Claims rejected: 1-11, 14-32
6. Claims objected to: none

C. Claims on Appeal: 1-11, 14-32

IV. STATUS OF AMENDMENTS

Appellant has not filed any amendments to the application subsequent to the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

A. Independent claim 1

Independent claim 1 recites a computing device comprising an application layer and an operating system layer having a first type of operating system and associated application program interfaces (APIs). (Page 4, lines 27-33 – Page 5, line 1, Fig. 1, 110, 116, and 120). The computing device also includes an interface module coupled between the application layer and the operating system layer, wherein the interface module receives program instructions from a program in the application layer written for a second type of operating system. (Page 5, lines 8-14, Page 6, lines 28-30; Page 9, lines 25-32; Fig. 1, 116, 128, and 120; Fig. 2, 218; Fig. 3, 318). The interface module processes the instructions through emulation, interpretation, translation, and conversion by directing the instructions to APIs that correctly execute the instructions. (Page 5, lines 1-4; Page 7, lines 7-8; Page 7 , lines 21-25; Page 8, lines 2-11; Page 8, lines 29-30; Page 14, lines 14-16). The interface module includes a discrete abstraction module having translation and conversion information therein and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein. (Page 10, lines 31-33 – Page 11, lines 1-4; Page 12, lines 1-17; Page 14, lines 23-27; Page 15, lines 22-25; Page 16, lines 19-23; Fig. 2, 224; Fig. 3, 324).

1. Claim 2 depends from independent claim 1 and recites the interface module includes an operating system emulation module for emulating a number of operating system functions. (Page 15, lines 22-25; Page 16, lines 30-32; Fig. 3, 318, 322).

2. Claim 3 depends from independent claim 1 and recites the interface module emulates operating system functions and network server functions. (Page 15, lines 11-19).

3. Claim 4 depends from independent claim 1 and recites the interface module emulates home location register functions. (Page 13, lines 19-28).

4. Claim 5 depends from independent claim 1 and recites the interface module emulates intelligent network server functions. (Page 13, lines 19-28).

5. Claim 6 depends from independent claim 1 and recites the interface module has portions for emulating the operating system functions and the network server functions in discrete modules located within the interface module. (Page 10, lines 31-33 – Page 11, lines 1-4; Page 12, lines 1-17; Page 14, lines 23-27; Page 15, lines 22-25; Page 16, lines 19-23; Fig. 3, 318, 322, 326).

6. Claim 7 depends from independent claim 1 and recites the interface module processes a program instruction by interpreting whether the instruction has to be processed further. (Page 7, lines 10-13 and 19-25).

7. Claim 8 depends from dependent claim 7 and recites the interface module converts a result received from the operating system layer such

that the converted result is in a format that the application program can use to execute the instruction. (Page 8, lines 29-33 – Page 9, line 1).

8. Claim 9 depends from dependent claim 7 and recites the application interface module translates the instruction received such that the operating system layer can execute the instruction. (Page 7, lines 4-8; Page 8, lines 13-27).

B. Independent claim 10

Independent claim 10 recites a system architecture comprising a computing device including an application layer having a home location register application. (Page 4, lines 25-31; Fig. 1, 116). The system architecture includes an operating system layer having a first type of operating system and an interface module to interface the home location register application designed for a second type of operating system with the first type of operating system through emulation, interpretation, translation, and conversion. (Page 4, lines 32-33 – Page 5, lines 1-6; Page 7, lines 6-8; Page 11, lines 14-20; Fig. 1, 120). The interface module includes a discrete abstraction module having translation and conversion information therein and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein. (Page 10, lines 31-33 – Page 11, lines 1-4; Page 12, lines 1-17; Page 14, lines 23-27; Page 15, lines 22-25; Page 16, lines 19-23; Fig. 2, 218, 222, and 224) The interface module also includes a connection for connecting the computing device to a publicly switched telephone network (PSTN). (Page 18, lines 25-28).

1. Claim 11 depends from independent claim 10 and recites the interface module has a number of modules to translate instructions between the operating system layer and the application layer. (Page 7, lines 4-8; Page 8, lines 13-27; Fig. 3, 322, 326, and 328).

2. Claim 12 was canceled.

3. Claim 13 was canceled.

4. Claim 14 depends from independent claim 10 and recites the operating system emulation module has translation and interpretation information therein. (Page 11, lines 26-30; Fig. 3, 322).

5. Claim 15 depends from independent claim 10 and recites the system architecture further includes an operating system emulation module to direct an instruction from the home location register application to an application program interface. (Page 13, lines 19-33 – Page 14, lines 1-6; Fig. 3, 322).

6. Claim 16 depends from independent claim 10 and recites the system architecture further includes a number of component modules that can interface between an application designed for a second type of operating system and the operating system layer having a first type of operating system. (Page 16, lines 25-33; Fig. 3, 328).

C. Independent claim 17

Independent claim 17 recites a method of executing an application comprising providing an application configured for an operating system and communicating instructions from the application to an interface module. (Page 5, lines 28-33; Fig. 1, 110, 116, 118, and 120; Fig. 2, 218; Fig. 3, 318). The interface

module includes a discrete abstraction module having translation and conversion information therein and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein. (Page 10, lines 31-33 – Page 11, lines 1-4; Page 11, lines 26-30; Page 12, lines 1-17; Page 14, lines 23-27; Page 15, lines 22-25; Page 16, lines 19-23; Fig. 2, 218, 222, and 224). The method also includes processing the instructions with the interface module through emulation, interpretation, translation, and conversion to function with a different operating system. (Page 5, lines 1-4; Page 7, lines 7-8; Page 7 , lines 21-25; Page 8, lines 2-11; Page 8, lines 29-30; Page 14, lines 14-16).

1. Claim 18 depends from independent claim 17 and recites processing the instructions from the application with the interface module includes using a list of instructions to be processed. (Page 20, lines 31-33 – Page 21, lines 1-3; Fig. 1, 118, Fig. 2, 218; Fig. 3, 318).

2. Claim 19 depends from independent claim 17 and recites the application is configured for a Linux based operating system. (Page 5, lines 24-26; Page 20, lines 27-29).

3. Claim 20 depends from independent claim 17 and recites the application is configured for a Windows based operating system. (Page 5, lines 24-26; Page 20, lines 27-29).

4. Claim 21 depends from independent claim 17 and recites the application is configured for a UNIX based operating system. (Page 5, lines 24-26; Page 20, lines 27-29).

5. Claim 22 depends from independent claim 17 and recites the method further includes identifying instructions to be translated by the interface module. (Page 7, lines 4-8; Page 8, lines 13-27; Fig. 2, 218; Fig. 3, 318).

D. Independent claim 23

Independent claim 23 recites a method of executing an application configured for a platform having first type of operating system on a platform having a second type of operating system comprising communicating instructions from the application to an interface module. (Page 5, lines 28-33; Fig. 1, 118, 120). The application is configured for a first type of operating system. (Page 5, lines 28-33). The method also includes interpreting the instructions from the application with the interface module through emulation, interpretation, translation, and conversion and communicating the instructions from the interface module to an operating system that is the second type of operating system. (Page 5, lines 1-4; Page 7, lines 7-8; Page 7 , lines 21-25; Page 8, lines 2-11; Page 8, lines 29-30; Page 14, lines 14-16; Fig. 1, 118; Fig. 2, 218; Fig. 3, 318). The interface module includes a discrete abstraction module having translation and conversion information therein and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein. (Page 10, lines 31-33 – Page 11, lines 1-4; Page 11, lines 26-30; Page 12, lines 1-17; Page 14, lines 23-27; Page 15, lines 22-25; Page 16, lines 19-23; Fig. 2, 224; Fig. 3, 322, 324).

1. Claim 24 depends from independent claim 23 and recites communicating instructions from the application to an interface module includes

communicating instructions to an operating system emulation module within the interface module. (Page 17, lines 1-6; Fig. 2, 218; Fig. 3, 318, 322).

2. Claim 25 depends from dependent claim 24 and recites interpreting the instructions includes directing an instruction from the operating system emulation module to an application program interface. (Page 14, lines 14-27; Page 15, lines 22-25; Fig. 2, 218; Fig. 3, 318, 322).

3. Claim 26 depends from independent claim 23 and recites communicating instructions from the application to an interface module includes communicating instructions to a network server emulation module within the interface module. (Page 15, lines 27-29; Fig. 2, 218; Fig. 3, 318, 326).

4. Claim 27 depends from independent claim 23 and recites interpreting the instructions includes translating an instruction configured for the first type of operating system to an instruction configured for the second type of operating system. (Page 4, lines 32-33 – Page 5, lines 1-6).

5. Claim 28 depends from independent claim 23 and recites interpreting the instructions includes converting a result configured for the second type of operating system to a result configured for the first type of operating system. (Page 8, lines 29-33 – Page 9, line 1).

E. Independent claim 29

Independent claim 29 recites a computer readable medium having a set of computer executable instructions thereon for causing a device to perform a method, comprising communicating instructions from a telecommunications application to an interface module, the telecommunication application configured for a first type of

operating system. (Page 4, lines 25-32; Fig. 1, 118). The method performed by the device also includes processing the instructions from the telecommunication application with the interface module through emulation, interpretation, translation, and conversion and communicating the instructions from the interface module to an operating system that is a second type of operating system. (Page 5, lines 1-4; Page 7, lines 7-8; Page 7 , lines 21-25; Page 8, lines 2-11; Page 8, lines 29-30; Page 14, lines 14-16; Fig. 1, 118; Fig. 2, 218; Fig. 3, 318). The interface module includes a discrete abstraction module having translation and conversion information therein and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein. (Page 10, lines 31-33 – Page 11, lines 1-4; Page 11, lines 26-30; Page 12, lines 1-17; Page 14, lines 23-27; Page 15, lines 22-25; Page 16, lines 19-23; Fig. 2, 218, 224; Fig. 3, 318, 322, and 324).

1. Claim 30 depends from independent claim 29 and recites communicating instructions from an application to an interface module includes communicating to an abstraction module within the interface module. (Page 11, lines 26-30; Fig. 2, 218, 224).

2. Claim 31 depends from independent claim 29 and recites communicating instructions from an application to an interface module includes communicating instructions to a component module within the interface module. (Page 16, lines 25-32; Fig. 3, 318, 328).

3. Claim 32 depends from independent claim 29 and recites the method further includes identifying instructions to be converted by the interface module. (Page 7, lines 10-13 and 19-25; Fig. 1, 118; Fig. 2, 218; Fig. 3, 318).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether or not claims 1-3, 5-9 and 17-28 are unpatentable under 35 USC § 103(a) as being unpatentable over Nowlin, Jr. et al. (U.S. Patent No. 6,484,309) in view of Hutchison et al. (U.S. Patent No. 6,651,123).

B. Whether or not claims 4, 10, 11, 14-16 and 29-32 are unpatentable under 35 USC § 103(a) as being unpatentable over Nowlin, Jr. et al. (U.S. Patent No. 6,484,309) in view of Hutchison et al. (U.S. Patent No. 6,651,123) and further in view of Fletcher et al. (U.S. Statutory Invention Registration No. H1,921).

VII. ARGUMENT

A. Arguments against the rejections under 103(a) over the Nowlin, Jr. '309 reference in view of the Hutchison '123 reference.

1. Arguments regarding claims 1, 17, and 23.

Applicant respectfully submits that Nowlin, Jr. and Hutchison, independently or in combination, do not describe, teach, or suggest all of the elements and limitations of independent claims 1, 17, and 23. For example, Nowlin, Jr. does not appear to teach an interface module that includes:

> a discrete abstraction module having translation and conversion information therein; and
>
> a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein

as provided in independent claims 1, 17, and 23.

The Nowlin, Jr. reference appears to teach a method in which a translation layer is created on a non-Windows® CE computer system to operate software on a Windows® CE computer system where the translation layer communicates to the different computer systems by using the calling convention of each computer system. (Col. 8, lines 6-13). The Nowlin, Jr. reference appears to teach a translation layer that creates a surrogate set of kernel files that allow for communication between the two computer systems. (Col. 3, lines 4-18).

The Examiner states at page 17 of the Office Action dated March 20, 2008 that "the translation layer abstracts the communications between the application layer and the operating system layer by hiding the implementation details of a particular set of functionality in the operating system layer and allowing for translation/conversion of application request/call such that the operating system layer would understand the application request/call."

The Nowlin, Jr. reference appears to teach, "a collaborating set 23 of kernel files contained in the conversion layer 22 handles translations between ASCII and Unicode and Unicode and ASCII as necessary." (Col. 3, lines 26-28). The conversion layer that handles the translation in the Nowlin, Jr. reference does not teach a "discrete abstraction module" or "a discrete operation system module", as recited in independent claims 1, 17, and 23, to perform the translation to create the collaborating set of kernel files.

Although the Nowlin, Jr. reference appears to allow for software programs written for operation on a non-Windows® CE computer system to operate on a Windows® CE computer system by creating a surrogate set of kernel files, the

Nowlin, Jr. reference does not appear to teach an interface module that includes "a discrete abstraction module having translation and conversion information therein" and "a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein," as provided in independent claims 1, 17, and 23. Applicant submits that the translation layer created in Nowlin, Jr., which uses the calling convention of two operation systems for communication between the two operation systems, does not appear to describe, teach, or suggest, a discrete abstraction module in communication with a discrete operating system emulation module as combined with other claim limitations in claims 1, 17, and 23.

Also, the translation layer created in Nowlin, Jr. that uses the calling convention of two operation systems for communication between the two operation systems does not appear to teach or suggest the emulation, interpretation, translation, and conversion as combined with other claim limitations in claim 10. Nowlin, Jr. merely translates ASCII strings from Windows 95 applications to Unicode strings for use on a Windows® CE system (Col. 3, lines 19-22), but does not interface an application through emulation, interpretation, translation, and conversion, as recited in claims 1, 17, and 23.

The Examiner cites the Hutchison reference as teaching an interface module that receives program instructions from the application layer and processes the instruction through emulation with "a discrete operating system emulation module in communication with a discrete abstraction module and having interpretation information therein." (Office Action, Pages 3, 6, and 8). However, from

Applicant's review of the Hutchison reference, the reference does not cure the deficiencies of Nowlin, Jr.

The Hutchison reference appears to describe a system that uses a file locking emulator between the application program and an operating system. (Col. 1, lines 64-67). The file lock emulator described in Hutchison appears to allow compatible requests from the application program to pass to the operating system, while incompatible files are returned with an error and corrected (Col. 10, lines 19-38), but Hutchison uses a file lock emulator to do this and not teach a "discrete abstraction module" or a "discrete operating system emulation module."

The Hutchison reference does not describe, teach, or suggest an interface module that includes "a discrete abstraction module having translation and conversion information therein" and "a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein," as provided in Applicant's independent claims 1, 17, and 23.

As such, the Applicant respectfully submits that each and every limitation of Applicant's independent claims 1, 17, and 23 are not described, taught, or suggested by the Nowlin, Jr. and Hutchison references, alone or in combination. Accordingly, Applicant respectfully requests reconsideration and withdrawal of the §103 rejection of independent claim 1, 17, and 23.

2. Arguments regarding claims 2-3, 5-9, 18-22, and 24-28.

Claims 2-3 and 5-9 depend from independent claim 1. Applicant respectfully submits that independent claim 1 is in condition for allowance. Accordingly, Applicant requests reconsideration and withdrawal of the rejection of

claims 2-3 and 5-9 that depend therefrom.

Claims 18-22 depend from independent claim 17. Applicant respectfully submits that independent claim 17 is in condition for allowance. Accordingly, Applicant requests reconsideration and withdrawal of the rejection of claims 18-22 that depend therefrom.

Claims 24-28 depend from independent claim 23. Applicant respectfully submits that independent claim 23 is in condition for allowance. Accordingly, Applicant requests reconsideration and withdrawal of the rejection of claims 24-28 that depend therefrom.

B. Arguments against the rejections under 103(a) over the Nowlin, Jr. '309 reference in view of the Hutchison '123 reference and in further view the Fletcher H1-921 reference.

1. Arguments regarding claims 10 and 29.

Applicant respectfully submits that Nowlin, Jr., Hutchison, and Fletcher, independently or in combination, do not describe, teach, or suggest all of the elements and limitations of independent claims 10 and 29.

For example, Nowlin, Jr. does not appear to teach an interface module that includes:

> a discrete abstraction module having translation and conversion information therein; and
> a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein

as provided in independent claims 10 and 29.

The Nowlin, Jr. reference appears to teach a method in which a translation layer is created on a non-Windows® CE computer system to operate software on a Windows® CE computer system where the translation layer communicates to the different computer systems by using the calling convention of each computer system. (Col. 8, lines 6-13). The Nowlin, Jr. reference appears to teach a translation layer that creates a surrogate set of kernel files that allow for communication between the two computer systems. (Col. 3, lines 4-18).

The Examiner states at page 17 of the Office Action dated March 20, 2008 that "the translation layer abstracts the communications between the application layer and the operating system layer by hiding the implementation details of a particular set of functionality in the operating system layer and allowing for translation/conversion of application request/call such that the operating system layer would understand the application request/call."

The Nowlin, Jr. reference appears to teach, "a collaborating set 23 of kernel files contained in the conversion layer 22 handles translations between ASCII and Unicode and Unicode and ASCII as necessary." (Col. 3, lines 26-28). The conversion layer that handles the translation in the Nowlin, Jr. reference does not teach a "discrete abstraction module" or "a discrete operation system module", as recited in independent claims 10 and 29, to perform the translation to create the collaborating set of kernel files.

Although the Nowlin, Jr. reference appears to allow for software programs written for operation on a non-Windows® CE computer system to operate on a Windows® CE computer system by creating a surrogate set of kernel files, the

Nowlin, Jr. reference does not appear to teach an interface module that includes "a discrete abstraction module having translation and conversion information therein" and "a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein," as provided in independent claims 10 and 29. Applicant submits that the translation layer created in Nowlin, Jr., which uses the calling convention of two operation systems for communication between the two operation systems, does not appear to describe, teach, or suggest, a discrete abstraction module in communication with a discrete operating system emulation module as combined with other claim limitations in claims 10 and 29.

Also, the translation layer created in Nowlin, Jr. that uses the calling convention of two operation systems for communication between the two operation systems does not appear to teach or suggest the emulation, interpretation, translation, and conversion as combined with other claim limitations in claims 10 and 29. Nowlin, Jr. merely translates ASCII strings from Windows 95 applications to Unicode strings for use on a Windows® CE system (Col. 3, lines 19-22), but does not interface an application through emulation, interpretation, translation, and conversion, as recited in claims 10 and 29.

The Examiner cites the Hutchison reference as teaching an interface module that receives program instructions from the application layer and processes the instruction through emulation with "a discrete operating system emulation module in communication with a discrete abstraction module and having interpretation information therein." (Office Action, Pages 12 and 14-15). However, from

Applicant's review of the Hutchison reference, the reference does not cure the deficiencies of Nowlin, Jr.

As described above, the Hutchison reference appears to describe a system that uses a file locking emulator between the application program and an operating system. (Col. 1, lines 64-67). The file lock emulator described in Hutchison appears to allow compatible requests from the application program to pass to the operating system, while incompatible files are returned with an error and corrected (Col. 10, lines 19-38), but Hutchison uses a file lock emulator to do this and not teach a "discrete abstraction module" or a "discrete operating system emulation module."

The Hutchison reference does not describe, teach, or suggest an interface module that includes "a discrete abstraction module having translation and conversion information therein" and "a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein," as provided in Applicant's independent claims 10 and 29.

Also, from Applicant's review of the Fletcher reference, the reference does not cure the deficiencies of the Nowlin, Jr. and Hutchison references. That is, Fletcher appears to teach a computing device including an application layer having a home location register application thereon, as stated by the Examiner, and the Fletcher reference does not appear to describe, teach, or suggest an interface module that includes "a discrete abstraction module having translation and conversion information therein" and "a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein," as provided in Applicant's independent claims 10 and 29.

As such, the Nowlin, Jr., Hutchison, and Fletcher references do not appear to describe, teach, or suggest, either individually or in combination, each and every element and limitation in Applicant's independent claims 10 and 29. Accordingly, Applicant respectfully requests reconsideration and withdrawal of the § 103 rejection of independent claims 10 and 29.

2. Arguments regarding claims 11, 14-16, and 30-32.

Claims 11 and 14-16 depend from independent claim 10. Applicant respectfully submits that independent claim 10 is in condition for allowance. Accordingly, Applicant requests reconsideration and withdrawal of the rejection of claims 11 and 14-16 that depend therefrom.

Claims 30-32 depend from independent claim 29. Applicant respectfully submits that independent claim 29 is in condition for allowance. Accordingly, Applicant requests reconsideration and withdrawal of the rejection of claims 30-32 that depend therefrom.

CONCLUSION

Appellant respectfully submits that the claims are in condition for allowance and notification to that effect is earnestly requested. The Examiner and/or members of the Board are invited to telephone Appellant's attorney Edward J. Brooks III at (612) 236-0120 to facilitate this appeal.

At any time during the pendency of this application, please charge any additional fees or credit overpayment to the Deposit Account No. 08-2025.

CERTIFICATE UNDER 37 C.F.R. §1.8: The undersigned hereby certifies that this correspondence is being transmitted to the United States Patent and Trademark Office facsimile number (571) 273-8300, on this 6 day of May, 2008.

Jennifer L. Vonhof
Name

Signature

Respectfully Submitted,
James A. Lamb, et al.

By their Representatives:

Brooks, Cameron & Huebsch, PLLC
1221 Nicollet Avenue, Suite 500
Minneapolis, MN 55403

Atty: Edward J. Brooks III
Reg. No.: 40,925

5/6/2008
Date:

VIII. CLAIMS APPENDIX

1. (Previously Presented) A computing device, comprising:

an application layer;

an operating system layer having a first type of operating system and associated application program interfaces (APIs); and

an interface module coupled between the application layer and the operating system layer, wherein the interface module receives program instructions from a program in the application layer written for a second type of operating system and processes the instructions through emulation, interpretation, translation, and conversion by directing the instructions to APIs that correctly execute the instructions; and

wherein the interface module includes:

a discrete abstraction module having translation and conversion information therein; and

a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein.

2. (Original) The computing device of claim 1, wherein the interface module includes an operating system emulation module for emulating a number of operating system functions.

3. (Original) The computing device of claim 1, wherein the interface module emulates operating system functions and network server functions.

4. (Original) The computing device of claim 1, wherein the interface module emulates home location register functions.

5. (Original) The computing device of claim 1, wherein the interface module emulates intelligent network server functions.

6. (Original) The computing device of claim 1, wherein the interface module has portions for emulating the operating system functions and the network server functions in discrete modules located within the interface module.

7. (Original) The computing device of claim 1, wherein the interface module processes a program instruction by interpreting whether the instruction has to be processed further.

8. (Original) The computing device of claim 7, wherein the interface module converts a result received from the operating system layer such that the converted result is in a format that the application program can use to execute the instruction.

9. (Original) The computing device of claim 7, wherein the application interface module translates the instruction received such that the operating system layer can execute the instruction.

10. (Previously Presented) A system architecture, comprising:

a computing device including:

an application layer having a home location register application thereon;

an operating system layer having a first type of operating system; and

an interface module to interface the home location register application designed for a second type of operating system with the first type of operating system through emulation, interpretation, translation, and conversion; and

wherein the interface module includes:

a discrete abstraction module having translation and conversion information therein; and

a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein; and

a connection for connecting the computing device to a publicly switched telephone network (PSTN).

11. (Original) The system architecture of claim 10, wherein the interface module has a number of modules to translate instructions between the operating system layer and the application layer.

12. (Canceled)

13. (Canceled)

14. (Previously Presented) The system architecture of claim 10, wherein the operating system emulation module has translation and interpretation information therein.

15. (Original) The system architecture of claim 10, wherein the system architecture further includes an operating system emulation module to direct an instruction from the home location register application to an application program interface.

16. (Original) The system architecture of claim 10, wherein the system architecture further includes a number of component modules that can interface between an application designed for a second type of operating system and the operating system layer having a first type of operating system.

17. (Previously Presented) A method of executing an application comprising:

providing an application configured for an operating system;

communicating instructions from the application to an interface module, wherein the interface module includes:

a discrete abstraction module having translation and conversion information therein; and

a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein; and

processing the instructions with the interface module through emulation, interpretation, translation, and conversion to function with a different operating system.

18. (Original) The method of claim 17, wherein processing the instructions from the application with the interface module includes using a list of instructions to be processed.

19. (Original) The method of claim 17, wherein the application is configured for a Linux based operating system.

20. (Original) The method of claim 17, wherein the application is configured for a Windows based operating system.

21. (Original) The method of claim 17, wherein the application is configured for a UNIX based operating system.

22. (Original) The method of claim 17, wherein the method further includes identifying instructions to be translated by the interface module.

23. (Previously Presented) A method of executing an application configured for a platform having first type of operating system on a platform having a second type of operating system comprising:

communicating instructions from the application to an interface module, the application configured for a first type of operating system;

interpreting the instructions from the application with the interface module through emulation, interpretation, translation, and conversion; and

communicating the instructions from the interface module to an operating system that is the second type of operating system; and

wherein the interface module includes:

a discrete abstraction module having translation and conversion information therein; and

a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein.

24. (Original) The method of claim 23, wherein communicating instructions from the application to an interface module includes communicating instructions to an operating system emulation module within the interface module.

25. (Original) The method of claim 24, wherein interpreting the instructions includes directing an instruction from the operating system emulation module to an application program interface.

26. (Original) The method of claim 23, wherein communicating instructions from the application to an interface module includes communicating instructions to a network server emulation module within the interface module.

27. (Original) The method of claim 23, wherein interpreting the instructions includes translating an instruction configured for the first type of operating system to an instruction configured for the second type of operating system.

28. (Original) The method of claim 23, wherein interpreting the instructions includes
converting a result configured for the second type of operating system to a result configured for the first type of operating system.

29. (Previously Presented) A computer readable medium having a set of computer executable instructions thereon for causing a device to perform a method, comprising:

communicating instructions from a telecommunications application to an interface module, the telecommunication application configured for a first type of operating system;

processing the instructions from the telecommunication application with the interface module through emulation, interpretation, translation, and conversion; and

communicating the instructions from the interface module to an operating system that is a second type of operating system;

wherein the interface module includes:

a discrete abstraction module having translation and conversion information therein; and

a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein.

30. (Original) The computer readable medium of claim 29, wherein communicating instructions from an application to an interface module includes communicating to an abstraction module within the interface module.

31. (Original) The computer readable medium of claim 29, wherein communicating instructions from an application to an interface module includes communicating instructions to a component module within the interface module.

32. (Original) The computer readable medium of claim 29, wherein the method further includes identifying instructions to be converted by the interface module.

IX. EVIDENCE APPENDIX

None

X. RELATED PROCEEDINGS APPENDIX

None